Question 1 — What is hypothesis testing in statistics?

Answer :

Hypothesis testing is a formal statistical procedure used to make decisions about a population parameter (for example, a population mean or proportion) based on sample data. The core idea is to evaluate whether observed sample data are consistent with a particular statement (the null hypothesis) about the population.

Key elements and steps:

1. **Formulate hypotheses**
   - **Null hypothesis (H_0):** A statement of no effect or no difference; it represents the default or status quo. Example: $H_0: \mu = \mu_0$.
   - **Alternative hypothesis (H_1 or H_a):** The statement the researcher wants to test — it reflects the presence of an effect, difference, or change. It can be one-sided (e.g., $\mu > \mu_0$) or two-sided ($\mu \neq \mu_0$).
2. **Choose a test statistic**
   - A test statistic is a function of the sample data that summarizes evidence about H_0. Common examples: Z-statistic, t-statistic, chi-square statistic, F-statistic, etc.
   - The statistic depends on the parameter being tested, the distributional assumptions, and whether population variance is known.
3. **Determine the sampling distribution under H_0**
   - Under the assumption that H_0 is true, the distribution of the test statistic is known or approximated (e.g., normal, t, chi-square). This allows computation of probabilities (p-values) for the observed statistic.
4. **Significance level (α)**
   - Pre-select the probability of incorrectly rejecting H_0 when H_0 is true (Type I error). Common values: 0.05, 0.01, 0.10.
5. **Compute p-value or critical region**
   - **p-value:** Probability, under H_0, of observing a test statistic as extreme (or more extreme) than the observed one. Small p-values indicate evidence against H_0.
   - **Critical region approach:** Compare test statistic to critical values determined by α.
6. **Decision**
   - If p-value $\leq \alpha$ (or test statistic falls into critical region), reject H_0 in favor of H_1.
   - Otherwise, fail to reject H_0 (do not accept H_0; simply insufficient evidence to reject it).
7. **Interpretation**
   - A rejection suggests the sample provides strong evidence that the parameter differs from the H_0 value.
   - Failing to reject does not prove H_0; it means sample data did not provide sufficient evidence against H_0.

Why hypothesis testing is useful

- It provides a structured, probabilistic approach to decision making with quantified error rates.
- Widely used across science, engineering, business, and social sciences to assess claims and experimental results.

Common assumptions & caveats

- Many tests assume independent observations and certain distributional forms (e.g., normality). When assumptions fail, alternative nonparametric tests or bootstrap methods may be preferable.
- The sample size affects which test is appropriate and whether approximations (e.g., CLT) are valid.
- Statistical significance ($p \leq \alpha$) is not the same as practical significance — effect sizes matter.

Question 2 — Null vs Alternative hypothesis

Answer :

Null hypothesis (H_0):

- Definition: H_0 is the default assumption or statement of no difference/effect. It is a precise statement about a population parameter that we assume true until evidence suggests otherwise.
- Purpose: Serves as the benchmark against which the alternative is tested.
- Examples:
    - One-sample mean: H_0: $\mu = 100$.
    - Proportion: H_0: $p = 0.5$.
    - Two-sample difference: H_0: $\mu_1 - \mu_2 = 0$.

Alternative hypothesis (H_1 or H_a):

- Definition: H_1 represents a claim that contradicts H_0; it describes the effect or difference the researcher suspects or wants to demonstrate.
- Forms:
    - Two-sided: H_1: parameter $\neq$ value (tests for any difference).
    - One-sided (right): H_1: parameter $>$ value (tests for an increase).
    - One-sided (left): H_1: parameter $<$ value (tests for a decrease).
- Example:
    - H_1: $\mu \neq 100$ (two-sided).
    - H_1: $p > 0.5$ (one-sided).

Key differences:

1. **Role in testing:** H_0 is assumed true initially; the test evaluates whether sample evidence is strong enough to reject H_0 in favor of H_1.
2. **Burden of proof:** Evidence must favor H_1 strongly enough (low p-value) to reject H_0. H_0 is not "proven" by failing to reject — we only fail to find evidence against it.
3. **Decision logic:** Statistical procedures are constructed to control the probability of incorrectly rejecting H_0 (Type I error) while trying to detect H_1 (power relates to Type II error).
4. **Directionality:** Alternative hypothesis determines whether the test is one-sided or two-sided; this affects critical regions and p-value calculations.

Practical considerations:

- State H_0 and H_1 before collecting data to avoid bias.
- Choose one- or two-sided alternative based on scientific question; one-sided tests require strong justification.

Question 3 — Significance level in hypothesis testing .

Answer :

Definition

- The significance level, denoted α, is the probability threshold set by the researcher for rejecting the null hypothesis when it is actually true. It quantifies the acceptable risk of making a Type I error (false positive).

Role and interpretation

- If $\alpha = 0.05$, we accept up to a 5% chance of incorrectly rejecting H_0 when H_0 is true.
- When computing a p-value: if $p \leq \alpha$, we reject H_0; if $p > \alpha$, we fail to reject H_0.
- α determines the critical values for the test statistic. For example, a two-sided Z-test with $\alpha = 0.05$ uses critical values ± 1.96.

Choosing α

- Conventional levels: 0.05, 0.01, 0.10. Selection depends on context:
  - Lower α (0.01) if consequences of Type I error are severe (e.g., approving a harmful drug).
  - Higher α (0.10) if missing an effect (Type II error) is more costly than a false positive.

Effect on decisions and trade-offs

- Decreasing α reduces Type I error probability but increases Type II error probability (β) unless sample size is increased.
- Power of a test = $1 - \beta$ (probability of correctly rejecting H_0 when H_1 is true). For fixed sample size, reducing α lowers power.

Example of use

- In a clinical trial, we might set $\alpha = 0.01$ to be conservative about claiming a new treatment is effective.
- In exploratory research, $\alpha = 0.10$ might be acceptable to detect potential effects for further study.

Summary

- α is a design choice reflecting acceptable false-positive risk.
- It must be set before data analysis.
- It affects critical values, p-value interpretation, and the balance between Type I and Type II errors.

Question 4 — Type I and Type II errors with examples

Answer:

Definitions

- **Type I error (false positive):** Rejecting H_0 when H_0 is actually true.
  - Probability of Type I error = α (the chosen significance level).
- **Type II error (false negative):** Failing to reject H_0 when H_1 is actually true.
  - Probability of Type II error = β.
  - Power of the test = $1 - \beta$ (probability of correctly rejecting H_0 when H_1 is true).

Illustrative examples

1. **Medical test example**
   - Scenario: Testing whether a new drug lowers blood pressure compared to placebo.
   - H_0: The drug has no effect (mean difference = 0).
   - H_1: The drug lowers blood pressure (mean difference < 0).
   - **Type I error:** Concluding the drug works (rejecting H_0) when it actually has no real effect. Consequence: approving an ineffective drug — possible harm to patients and wasted resources.
   - **Type II error:** Concluding the drug does not work (failing to reject H_0) when it truly does reduce blood pressure. Consequence: missing a beneficial treatment.
2. **Judicial analogy**

- o H₀: Defendant is innocent.
- o H₁: Defendant is guilty.
- o **Type I error:** Wrongful conviction (rejecting innocence when innocent).
- o **Type II error:** Wrongful acquittal (failing to convict when guilty).
3. **Quality control**
   - o H₀: A batch of products meets quality standard (defect rate acceptable).
   - o H₁: Batch does not meet quality standard (defect rate too high).
   - o **Type I error:** Rejecting a good batch — unnecessary waste or rework.
   - o **Type II error:** Accepting a bad batch — defective products reach customers.

Controlling the errors

- Lower α reduces Type I error but increases β (unless sample size increased).
- To reduce β (increase power), one can increase sample size, increase effect size, or use more sensitive measurement methods.
- The trade-off must be managed according to context (costs of false positives vs false negatives).

Numerical illustration (brief)

- If α = 0.05 and power = 0.8 for a particular alternative, then β = 0.2. This means a 5% chance of Type I error and a 20% chance of Type II error.

Question 5 — Difference between Z-test and T-test; when to use each

Answer :

Z-test

- Used when testing hypotheses about a population mean (or proportion) and the population standard deviation σ is known, or when the sample size is large enough that the sampling distribution of the sample mean is approximately normal by the central limit theorem and σ can be approximated.
- Test statistic (one-sample mean):

  $Z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$

  where σ is the known population standard deviation.

- The Z distribution (standard normal) is used to compute p-values and critical values.

T-test

- Used when the population standard deviation σ is unknown and the sample comes from a normally distributed population. The sample standard deviation s is used to estimate σ.
- For a one-sample t-test, the statistic is:

t=x⁻−μ0s/nt = \frac{\bar{x} - \mu_0}{s / \sqrt{n}}t=s/nx⁻−μ0

and it follows a Student's t-distribution with n−1n-1n−1 degrees of freedom.

- The t-distribution is wider (heavier tails) than the normal, reflecting additional uncertainty from estimating σ.

When to use which

1. **Z-test:** use when
   - σ is known (rare in practice), or
   - the sample size n is large (common rule-of-thumb $n \geq 30$) and you rely on the CLT; many practitioners use the z-approximation for large samples even when σ is unknown, substituting s for σ.
2. **T-test:** use when
   - σ is unknown and sample size is small (n < 30), and the data are approximately normally distributed.
   - Even for moderate sample sizes, t-test is safer when σ unknown because it accounts for extra uncertainty.

Examples

- Measuring whether the mean weight of apples differs from 150 g: if long-term σ known from past data use Z; otherwise use t-test.
- For proportions, z-tests using the normal approximation to the binomial are common when np and n(1−p) are large.

Practical note

- When in doubt and sample size is relatively small, prefer the t-test. Many statistical software packages automatically choose t-based procedures when σ is unknown.

Question 6 — Generate binomial distribution (n=10, p=0.5), plot histogram .

- I generated 10,000 repeated trials of a Binomial(n=10, p=0.5) experiment using `numpy.random.binomial` and plotted a histogram. The code and histogram were executed; the histogram image is shown in the output above.

Printed descriptive results from the run

- Observed mean (from simulation) ≈ 4.9749
- Observed variance ≈ 2.52387
- Theoretical mean = n·p = 5.0
- Theoretical variance = n·p·(1 − p) = 2.5

Explanation

- A binomial(n, p) distribution models the number of successes in n independent Bernoulli trials with success probability p.
- For n = 10 and p = 0.5, the distribution is symmetric with mean 5. The histogram from 10,000 simulated experiments shows the expected bell-shaped discrete distribution centered near 5, with most mass between 1 and 9 successes.
- The empirical mean and variance match closely with theoretical values (small differences due to simulation randomness).

Python code used (this is the same code that was run earlier in the output cell above):

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
n = 10
p = 0.5
trials = 10000
binom_samples = np.random.binomial(n, p, size=trials)
plt.hist(binom_samples, bins=range(n+2), edgecolor='black')
plt.title('Histogram of binomial(n=10, p=0.5) over 10,000 trials')
plt.xlabel('Number of successes')
plt.ylabel('Frequency')
plt.show()
```

Question 7 — Implement hypothesis testing using Z-statistics for the sample dataset .

Given sample_data

```
[49.1, 50.2, 51.0, 48.7, 50.5, 49.8, 50.3, 50.7, 50.2, 49.6,
 50.1, 49.9, 50.8, 50.4, 48.9, 50.6, 50.0, 49.7, 50.2, 49.5,
 50.1, 50.3, 50.4, 50.5, 50.0, 50.7, 49.3, 49.8, 50.2, 50.9,
 50.3, 50.4, 50.0, 49.7, 50.5, 49.9]
```

Objective

- Use Z-statistics to test H_0: $\mu = 50$ versus H_1: $\mu \neq 50$ (two-sided).

Approach and important assumption

- A true Z-test requires the population standard deviation σ to be known. In practice σ is rarely known.
- Common pragmatic approach: when sample size is large (rule of thumb n ≥ 30), we can use the Central Limit Theorem and approximate the sampling distribution of the sample mean as normal. Therefore, using the sample standard deviation s to compute the standard error and applying the normal (Z) approximation is acceptable as an approximation.
- Here, n = 36 (which is > 30), so we perform a Z-approximation test: compute $Z = (\bar{x} - \mu_0) / (s / \sqrt{n})$ and use the standard normal to obtain a p-value. We explicitly state this approximation.

Computed numeric results (from code run)

- Sample size n = 36
- Sample mean $\bar{x} \approx 50.0889$
- Sample standard deviation s ≈ 0.5365
- Standard error SE = s / $\sqrt{n} \approx 0.089423$
- Z-statistic ≈ 0.9940
- Two-sided p-value (normal approx) ≈ 0.32021

Decision at α = 0.05

- p-value = 0.32021 > 0.05 → Fail to reject H₀.
- Interpretation: There is insufficient evidence, at the 5% significance level, to conclude that the true mean differs from 50. The sample mean is close to 50 and the observed deviation is not statistically significant.

Remarks

- If one wanted an exact small-sample treatment (because σ is unknown), one would use the t-test with t-distribution and degrees of freedom n − 1 = 35. For large n, the t and z results will be very similar.
- Always state the assumptions (normality or large n, independence) and whether population variance is known or approximated.

Python code used):

```
import numpy as np
from math import sqrt
import scipy.stats as stats

arr = np.array(sample_data)
n_sample = len(arr)
sample_mean = arr.mean()
sample_std = arr.std(ddof=1)
se = sample_std / sqrt(n_sample)
z_stat = (sample_mean - 50.0) / se
p_value_two_sided = 2 * (1 - stats.norm.cdf(abs(z_stat)))
```

Question 8 — Simulate normal distribution, compute 95% CI for mean, and plot .

Task

- Simulate normal data and compute the 95% confidence interval for the mean. Plot the data.

Procedure used

- I simulated `n = 100` observations from Normal($\mu = 5.0$, $\sigma = 2.0$) (a representative example).
- Since σ is unknown in simulated samples, I computed the sample standard deviation `s` and used the standard error `s/√n`. For a 95% CI, one could use the t-critical value for n = 100 (df = 99) which is ~1.984; a common approximation uses z = 1.96. Because n is large, using z = 1.96 is acceptable; I used z = 1.96 for the reported CI (I also reported sample sd so you can see exact numbers).
- The histogram of the simulated sample was plotted and the sample mean and CI endpoints were drawn as vertical lines.

Numerical output (from the run)

- n = 100
- Sample mean $\approx$ 5.1212
- Sample standard deviation $\approx$ 1.7792
- 95% CI (approx using z = 1.96): [4.7724, 5.4699]

Interpretation

- The interval [4.7724, 5.4699] is an estimated range that, under repeated sampling, would contain the true mean ~95% of the time (subject to model assumptions and method).
- Because the true population mean was 5.0 in this simulation, the CI contains 5.0 (so this simulated dataset yields a CI consistent with the true mean).

Python code used (already run in execution above):

```
np.random.seed(1)
n_sim = 100
mu_true = 5.0
sigma_true = 2.0
data = np.random.normal(loc=mu_true, scale=sigma_true, size=n_sim)
sample_mean_8 = data.mean()
sample_std_8 = data.std(ddof=1)
se_8 = sample_std_8 / sqrt(n_sim)
z_95 = 1.96
```

```
ci_lower = sample_mean_8 - z_95 * se_8
ci_upper = sample_mean_8 + z_95 * se_8
```

Question 9 — Z-score function and visualization; interpret Z-scores

•

Task

- Write a Python function to calculate Z-scores from a dataset, visualize the standardized data using a histogram, and explain what the Z-scores represent.

Definition and explanation

- The Z-score (standard score) of an observation x is:

  $$z = \frac{x - \mu}{\sigma}$$

  where μ is the mean of the dataset (or the population mean) and σ is the standard deviation (population sd if standardizing to unit population scale).

- Interpretation:
  - A z-score indicates how many standard deviations an observation is from the mean.
  - $z = 0 \rightarrow$ observation equals the mean.
  - $z = +1 \rightarrow$ observation is one standard deviation above the mean.
  - $z = -2 \rightarrow$ observation is two standard deviations below the mean.
- When data are standardized, the transformed values have mean approximately 0 and standard deviation 1 (exactly 0 and 1 if population mean and population sd used for standardization; with sample estimates the empirical mean will be ~0 and sd ~1).

What the code did

- Implemented a `z_scores(dataset)` function that computes the standardized values using the dataset mean and population-style standard deviation (`ddof=0`).
- Used the simulated data from Q8 as example input, computed z-scores, printed first 10 values and summary statistics (mean $\approx$ 0, std $\approx$ 1), and plotted a histogram of z-scores.

Sample printed outputs from the run

- The first 10 z-scores (rounded): `[ 1.7667, -0.7596, -0.6651, -1.2806, 0.9092, -2.6686, 1.9027, -0.9284, 0.2920, -0.3502]`
- Mean of z-scores $\approx$ 0.0
- Standard deviation of z-scores $\approx$ 1.0

Interpretation

- The z-scores show the relative standing of each data point. For example, $z \approx 1.77$ means that the observation is about 1.77 standard deviations above the sample mean — fairly high but not extremely rare.
- If the original data are approximately normal, about 95% of z-scores will lie between -1.96 and $+1.96$, matching the familiar 95% rule.

Python function used (already run):

```
def z_scores(dataset):
    arr = np.array(dataset)
    mu = arr.mean()
    sigma = arr.std(ddof=0)
    if sigma == 0:
        return np.zeros_like(arr)
    return (arr - mu) / sigma
```